# 后台自动化

# 一、后台自动化简介

# 二、准备工作

## 1、安装和配置python环境

安装python版本：python-3.6.2-amd64.exe

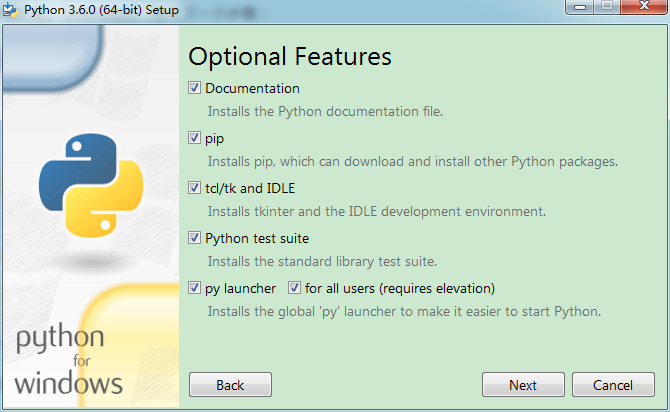
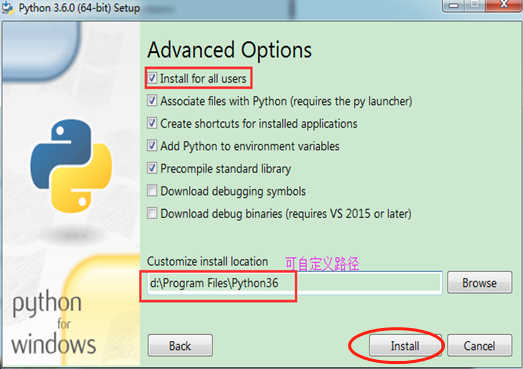
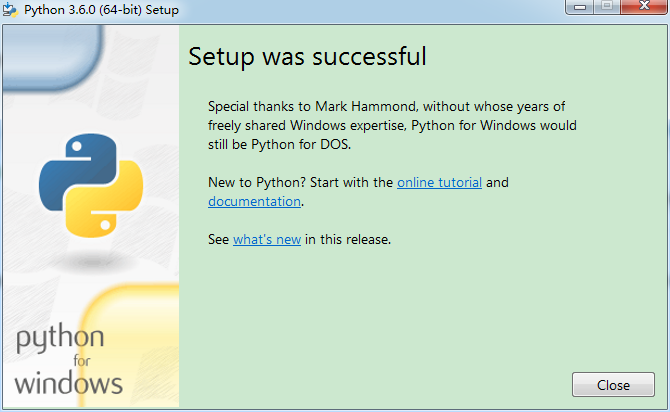
### [Python3.6.0安装](https://www.cnblogs.com/bbn0111/p/6612911.html)

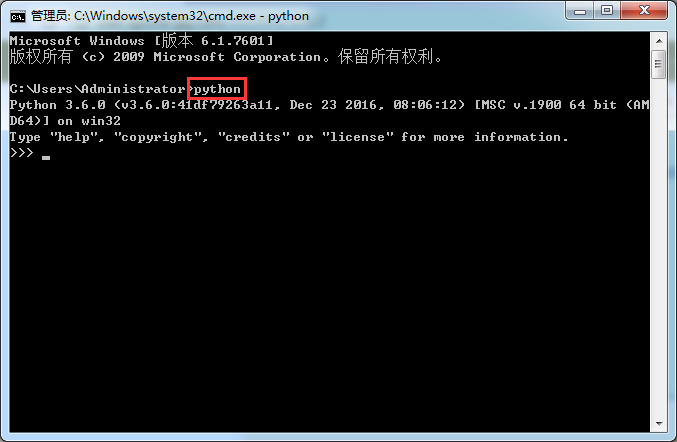
1.安装

具体详情请参考下图：

双击安装包，

勾选“add  python 3.6 to PATH”这样可以自动生成环境变量，选择“Customize installation”自定义安装。

2、验证是否安装成功  
    按“win+R”打开cmd命令，输入Python，看到如下界面，说明Python已经安装成功并正常运行。   
   

## 2、安装库

配置好python环境以后，在dos（win10 需要用管理员权限打开）窗口下，执行pip install -r requirements.txt

注意：requirements.txt文件在项目的根目录下，自行补充 requirements.txt 的具体路径

## 3、建表

使用ini.sql 下的sql建表，可以选择空的Oracle或者非空Oracle数据库

注意：sql里面的tablespace如果不是DATA01，自行修改

## 4、config.ini 修改

config.ini 在AutoTestV66\config 目录下

config.ini 需要修改的配置：

[DataSource] 、[Login] 、 [BackendPath] 、[Report] 、[Debug] 、[Mail]

#维护TESTCASE 数据的库信息**[DataSource]  
DatabaseConnectionUrl** = **172.31.6.234:1521/TESTDB  
DatabaseUser** = **PROMOTION  
DatabasePassword** = **PROMOTION**

#系统后台的访问信息，Url: 后台访问路径， User: 后台登录用户 ， Psw: 后台登录密码**[Backend]****Url** = **http://172.31.1.126/OK66/  
User** = **SZ001  
Psw** = **SZ001**

#自行选择是否定义，如果不填Path，报告生成在项目中的report/htmlreport 下**[Report]  
Path** =

#调试用例，mode = 0 表示执行用例表里面的全部用例，mode = 1，表示执行指定的用例，executeuser 表示执行指定的user用例，如果为空，表示执行没有指定user的用例**[Debug]  
Mode** = **1  
Execute\_User** = **xxx**

**[Mail]**# 0 表示关闭邮件发送，1 是开启**If\_Send** = **0**#Host,User,Psw这三个参数不需要修改**Host** = **smtp.tech-trans.com  
User** = **ttsz  
Psw** = **24945000**# 如果存在多个收件人，收件人地址之间用逗号隔开**Receivers** = **xxx.xxx@tech-trans.com**

注意：[Mail] config 中 黄色highlight的config 才需要修改

# 三、自动化启动

## 1、概况

**注意：以下所需要维护的数据都是先保存在Excel表格中，执行用例的时候都需要维护到数据库对应的表里面**

****

Excel中 五个sheet：分别是 TESTCASE 、MIXCASE 、CASEDEBUG 、 PAGELOCATION 、KEYWORD

TESTCASE ：维护测试用例 ，分别对应数据库中XF\_TESTCASE

MIXCASE ：组合测试用例 ，分别对应数据库中XF\_MIXCASE

CASEDEBUG ：调试测试用例 ，分别对应数据库中XF\_ CASEDEBUG

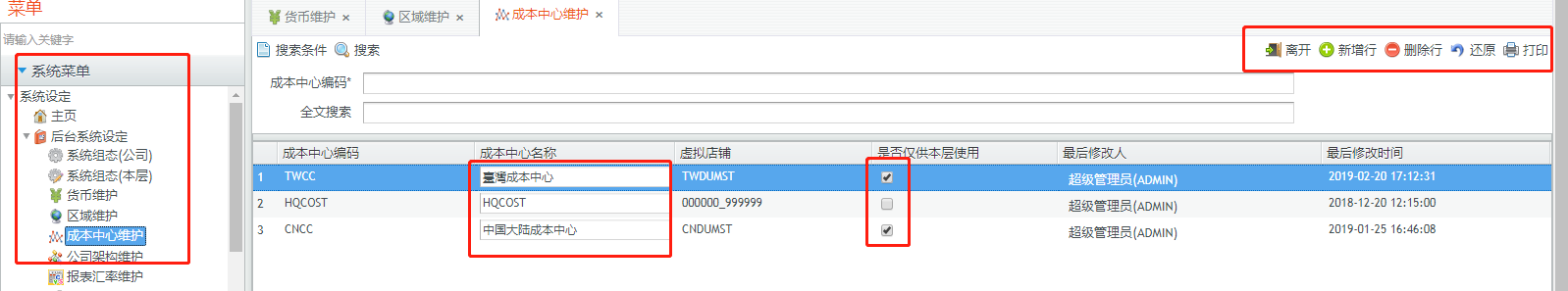
PAGELOCATION : 维护页面元素定位信息 ，分别对应数据库中XF\_ PAGELOCATION

## 2、维护和执行用例

### 2.1、维护元素定位信息

元素定位信息维护在**XF\_PAGELOCATION**

写用例前必须维护好页面元素ID



以上每一个按钮，每一个菜单，每一个输入框等，都需要维护对应的元素定位，元素定位就是为了告诉程序，操作对象的具体位置，所以每一个**元素定位**必须是**唯一**的

XF\_PAGELOCATION表如下所示：



**特别注意：XF\_LOCATIONID 必须是唯一，不允许重复**

XF\_FUNCTIONID : 指模块id，也就是系统菜单的对应的id

XF\_FUNCTIONNAME ：模块id对应名字

XF\_LOCATIONID ：元素id

XF\_LOCATIONNAME ：元素名字

XF\_LOCATIONDESC ：元素描述

XF\_LOCATION ：元素定位

#### 2.1.1常规元素维护

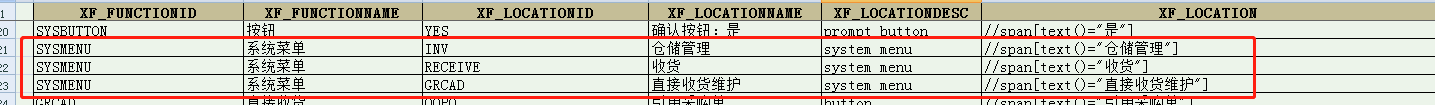
**XF\_PAGELOCATION 维护规则：**

**元素定位维护通常分为以下三种情况：**

##### A、菜单定位维护



示例：

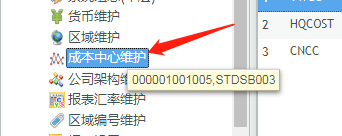


系统菜单元素定位维护规则：

XF\_FUNCTIONID统一为： SYSMENU

XF\_FUNCTIONNAME 统一为：系统菜单

**XF\_LOCATIONID** 取值：



如上方所示，菜单 成本中心维护 的locationid 为：STDSB003

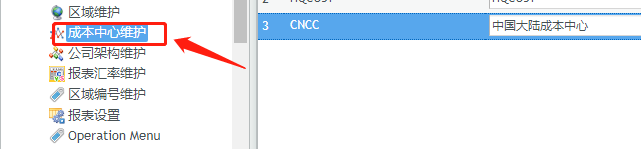
XF\_LOCATIONNAME: 为菜单名

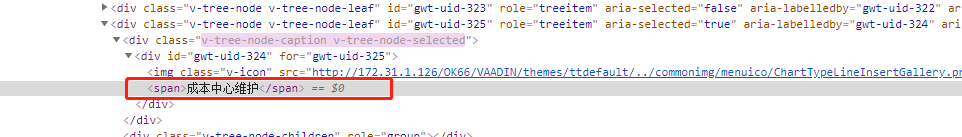
XF\_LOCATIONDESC: 菜单描述

**XF\_LOCATION**: 菜单定位信息

例如：

使用开发者工具找到定位信息





菜单定位暂时采用文本定位（以后需要修改为使用id定位）

以上 菜单： 成本中心维护： //span[text()="成本中心维护"]

文本定位格式为： //span[text()="XXX"] ，其中XXX是菜单名，注意，必须和菜单名字一致

Tips: //span[text()="XXX"] 是XPATH元素定位 （会在下面知识补充讲解）

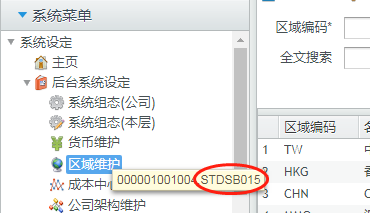
##### B、页面内元素定位维护



页面元素维护规则：

XF\_FUNCTIONID:

如图下所示：



XF\_FUNCTIONNAME:菜单名字

XF\_LOCATIONID:

例如某一个元素定位信息为：//\*[@id="XF\_REGIONCODE"]，因为locationid必须唯一，为了防止重复，可以加上functionid 信息，如STDSB015\_REGIONCODE ，如果id太长，自行缩减，例如可以改为S015\_REGIONCODE

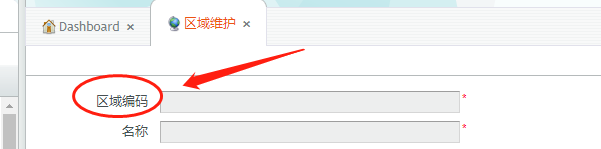
**注意：有些元素的locationid需要自行命名，如下所示：**

****

**命名的时候尽量起一些有意义的名字，如ADD，DEL 等**

XF\_LOCATIONNAME:元素的中文名称

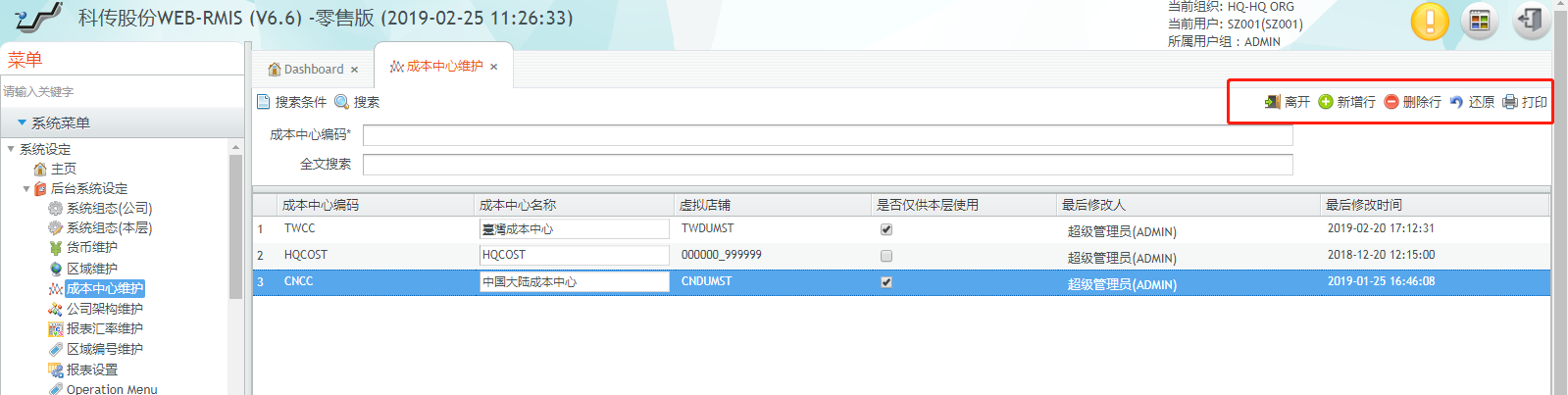
如图所示：

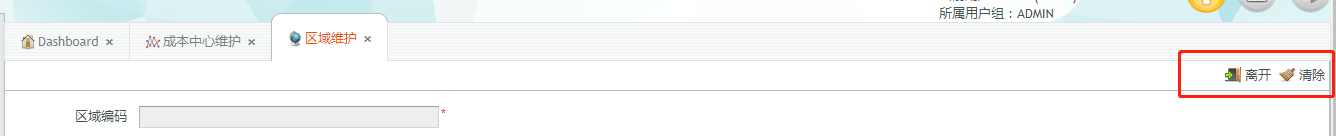


XF\_LOCATIONDESC：元素描述（有需要时自行添加描述）

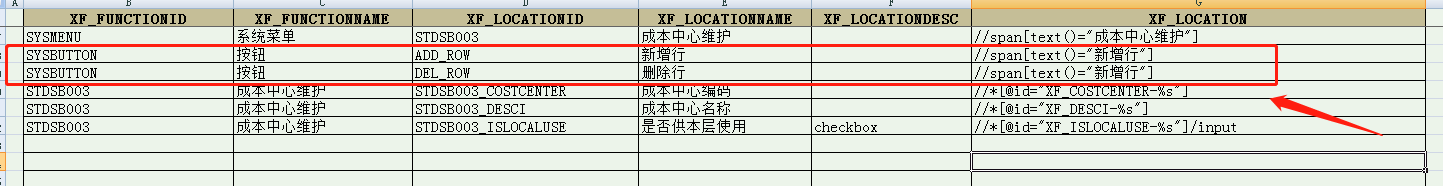
XF\_LOCATION:元素定位信息

##### C、按钮定位维护





像这些，页面上方的按钮，统一如下维护：



**按钮定位维护规则：**

XF\_FUNCTIONID: 统一维护 SYSBUTTON

XF\_FUNCTIONNAME: 统一维护 按钮

XF\_LOCATIONID: 自定义，尽量起一些有意义的名字，例如： 按钮：删除行 ，locationid 可以命名为ADD\_ROW

XF\_LOCATIONNAME : 按钮名字

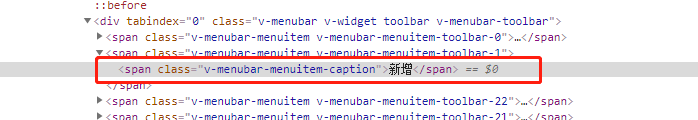
XF\_LOCATIONDESC ：按钮描述

XF\_LOCATION : 按钮定位

如:



使用开发者工具查到此按钮定位信息如下：

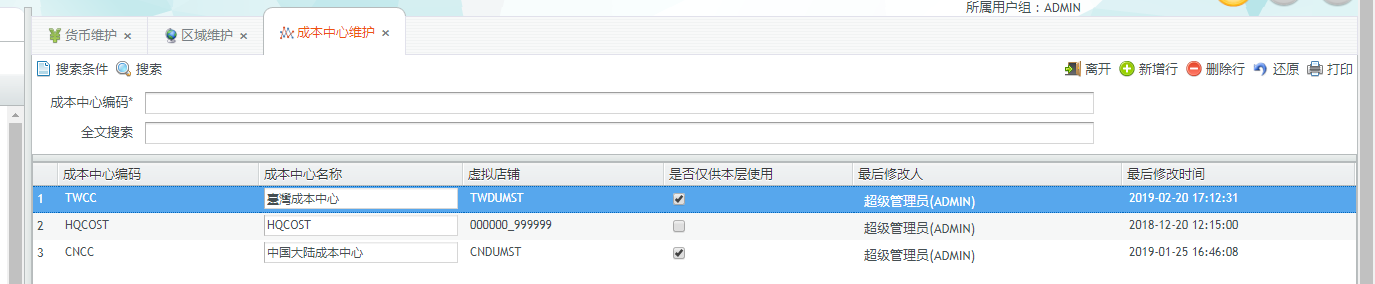


按钮没有元素id，所以统一使用文本定位的方法:

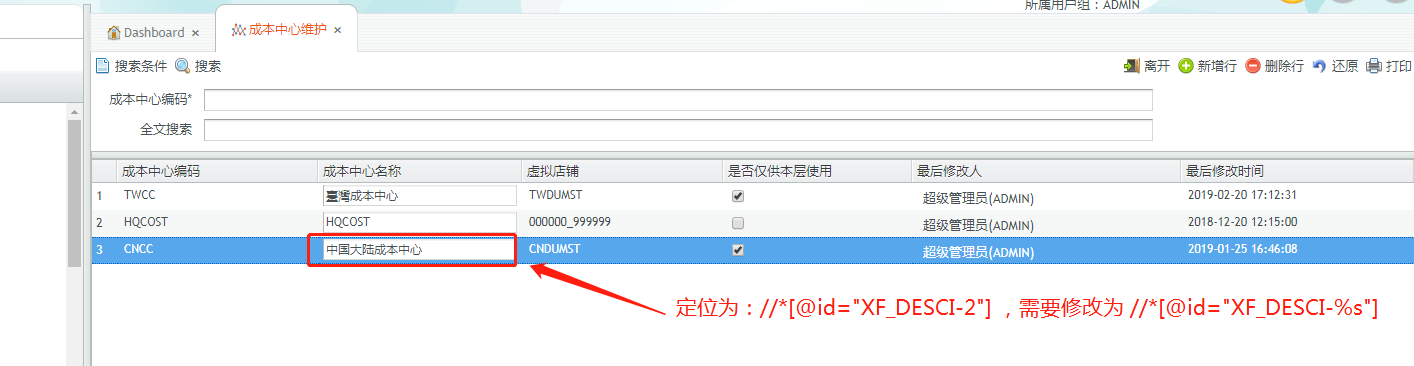
新增按钮定位： //span[text()="新增"]

#### 2.1.2特殊元素维护

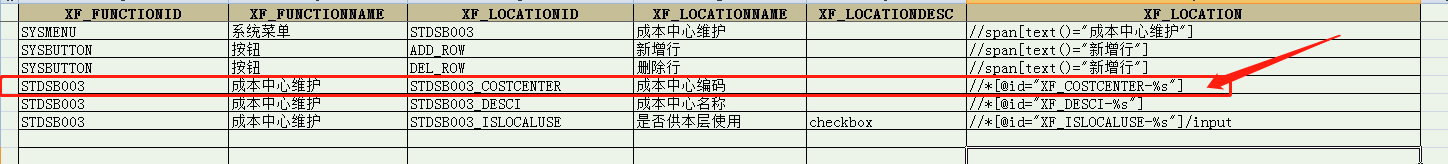
##### 1）、添加行，可以添加无限行，元素有无限多



**例如：**

****

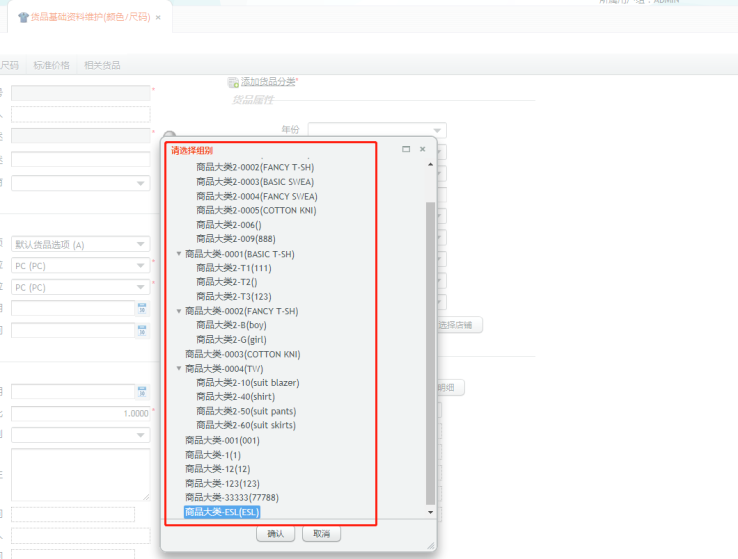
**表中维护如下：**

****

**注意：**

像以上多行元素，只需要维护一行即可

##### 2）树状菜单



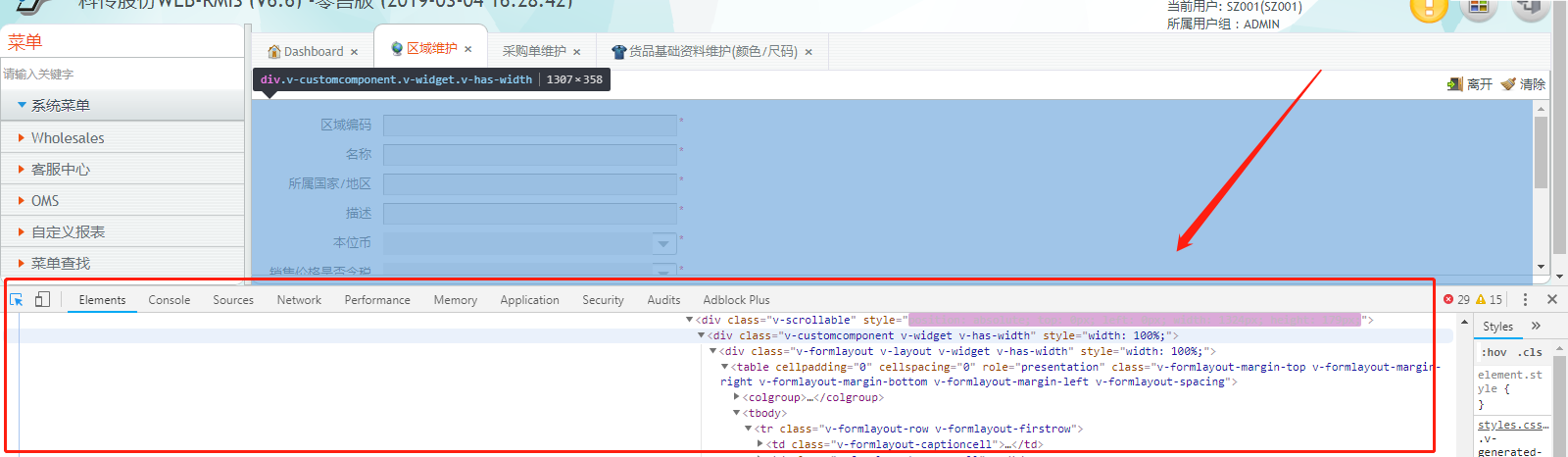
树状菜单里面的元素不需要维护，因为像这一类的菜单不是固定不变的，是可以动态修改的，所以不维护

#### 知识补充：定位补充

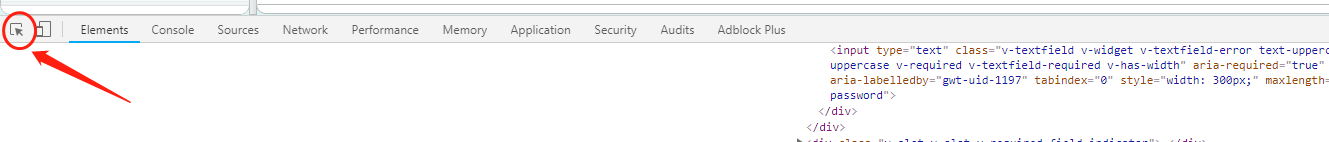
##### 1）定位的基本操作

**以下定位操作均在Chrome浏览器下进行**

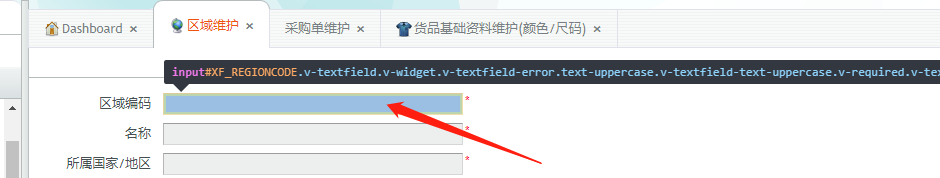
I、一般情况下，页面元素都能用开发者工具直接定位，需要定位时，点击键盘按钮 F12，出现以下界面时，即可定位元素



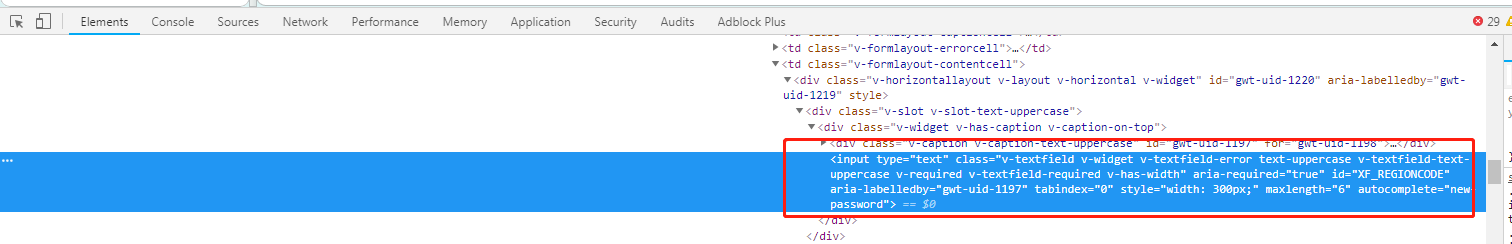
II、需要定位某个元素时，要首先点击以下位置，箭头变为蓝色时即可开始定位



III、将鼠标移动到需要定位的位置，如下所示：

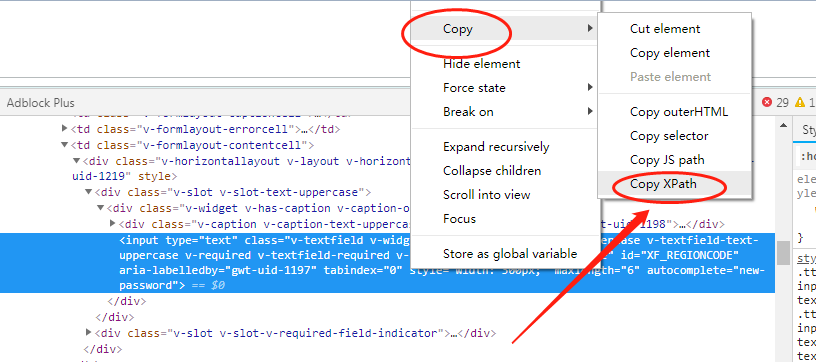


如上，定位一个输入框，移动到输入框的位置后，鼠标左键点击一下，即可定位该输入框



蓝色highlight信息就是响应的定位信息

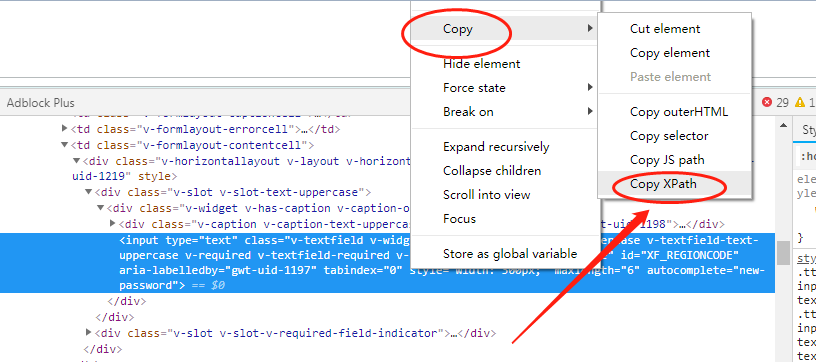
IV、右击蓝色highlight的任意一个位置，copy xpath 信息，该信息就是元素定位信息



如图上，copy的xpath信息为 //\*[@id="XF\_REGIONCODE"]

##### 2）元素定位的普通情况

一般元素都可以用copy xpath 的方式来获取定位信息

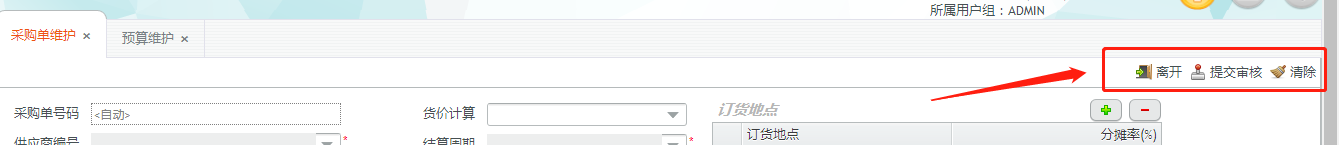


##### 3）无法直接定位的元素

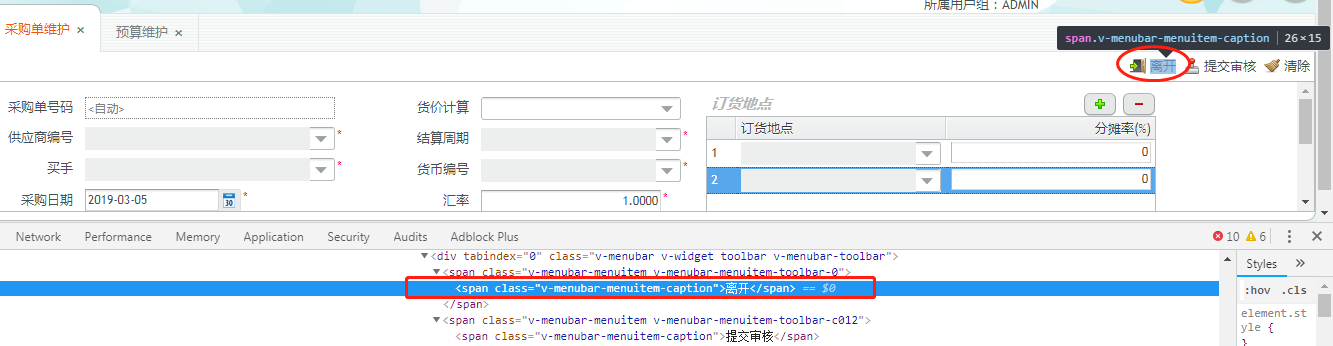
总的来说，元素定位，就是需要找到需要定位的元素的唯一路径，不能直接定位到的，都需要间接的定位

###### A、定位按钮

* 有文本标识的按钮



定位“离开”按钮



可以看出来 span标签里面没有id，需要看一下span 里面的属性 class

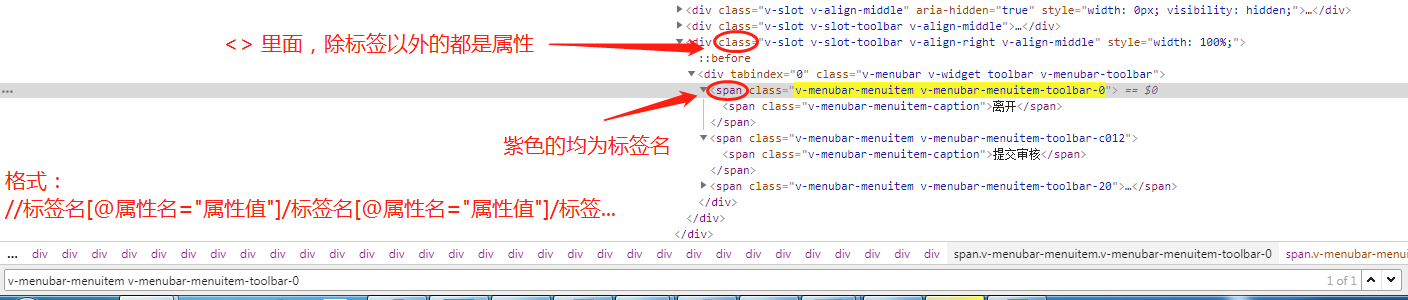


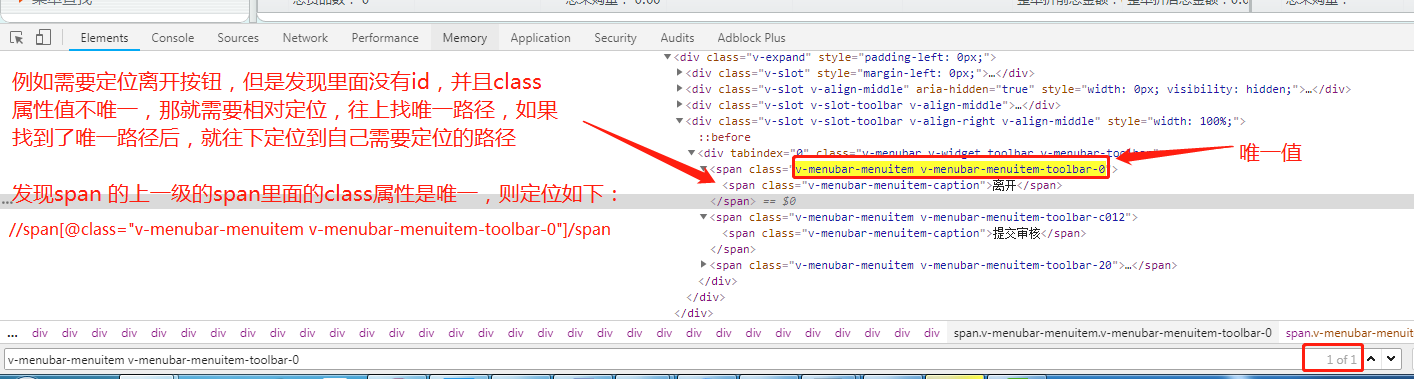
通过查找观察，span标签的上一级标签span里面的属性是唯一的



**【知识补充】**

**元素定位信息书写格式**：

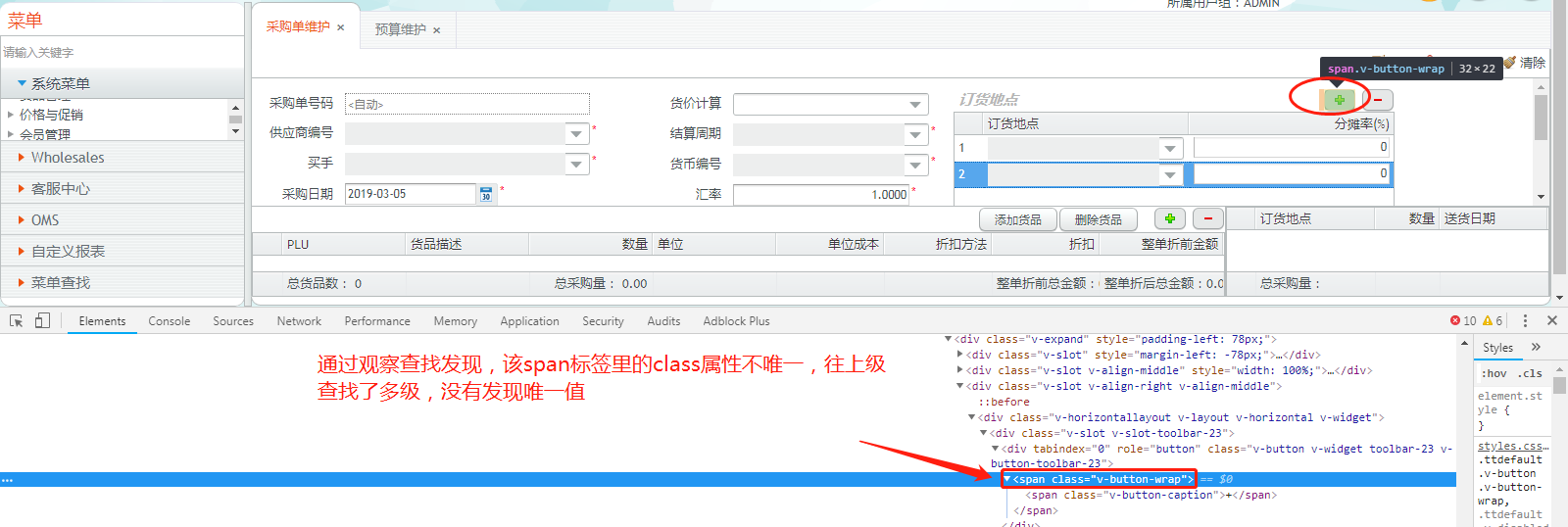




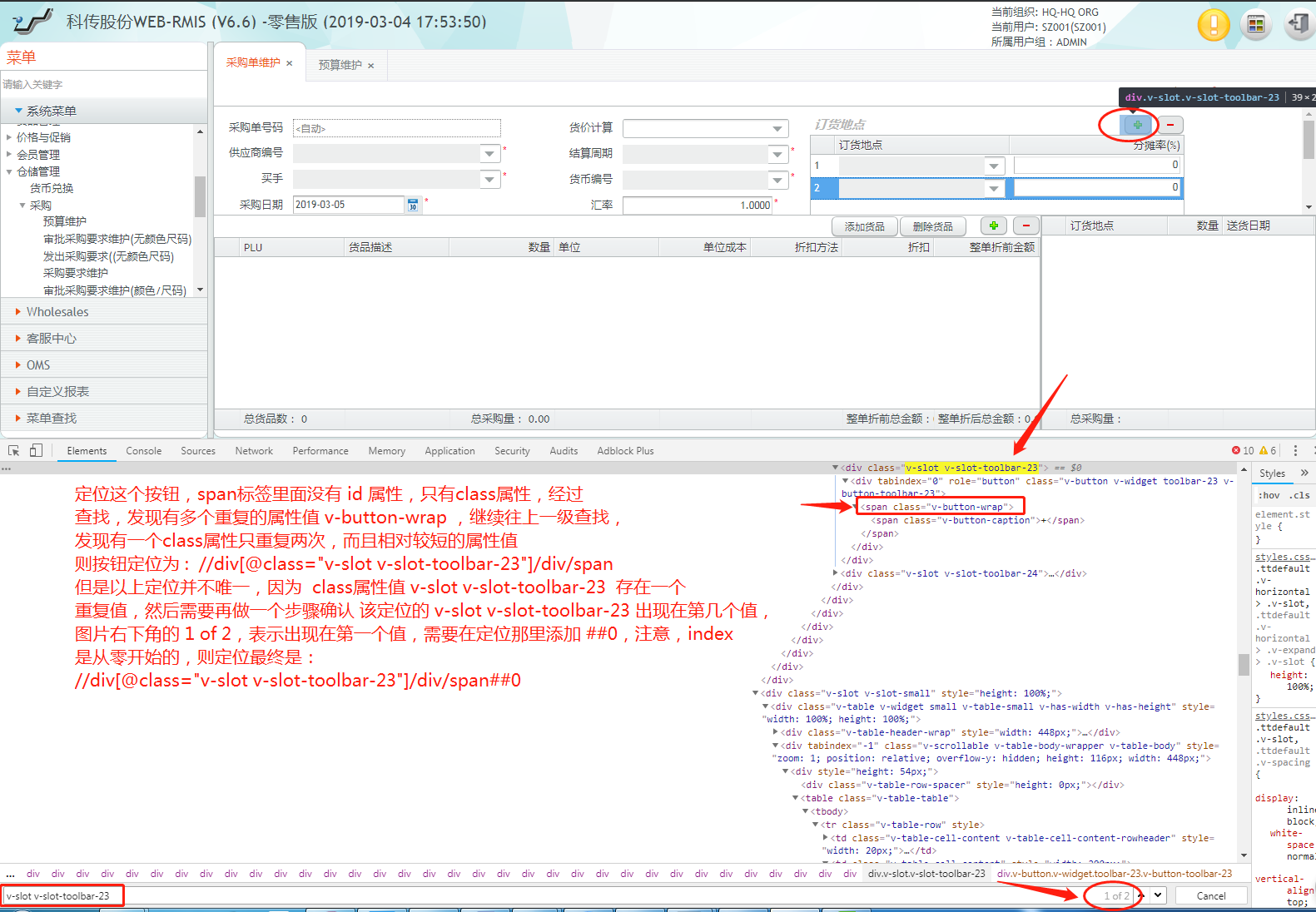
* **没有文本标识的按钮**

**这种定位需要特殊处理**

**如图按钮：**

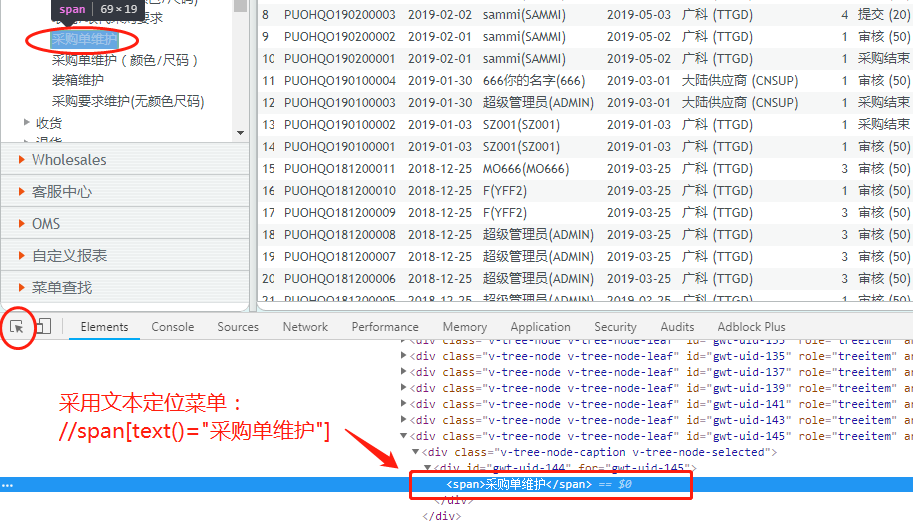
****

这种没有办法直接找到唯一路径的需要经过特殊处理



###### B、定位系统菜单

系统菜单暂时使用文本定位，需要等待开发人员给系统菜单赋值唯一id再修改



**知识点补充**：

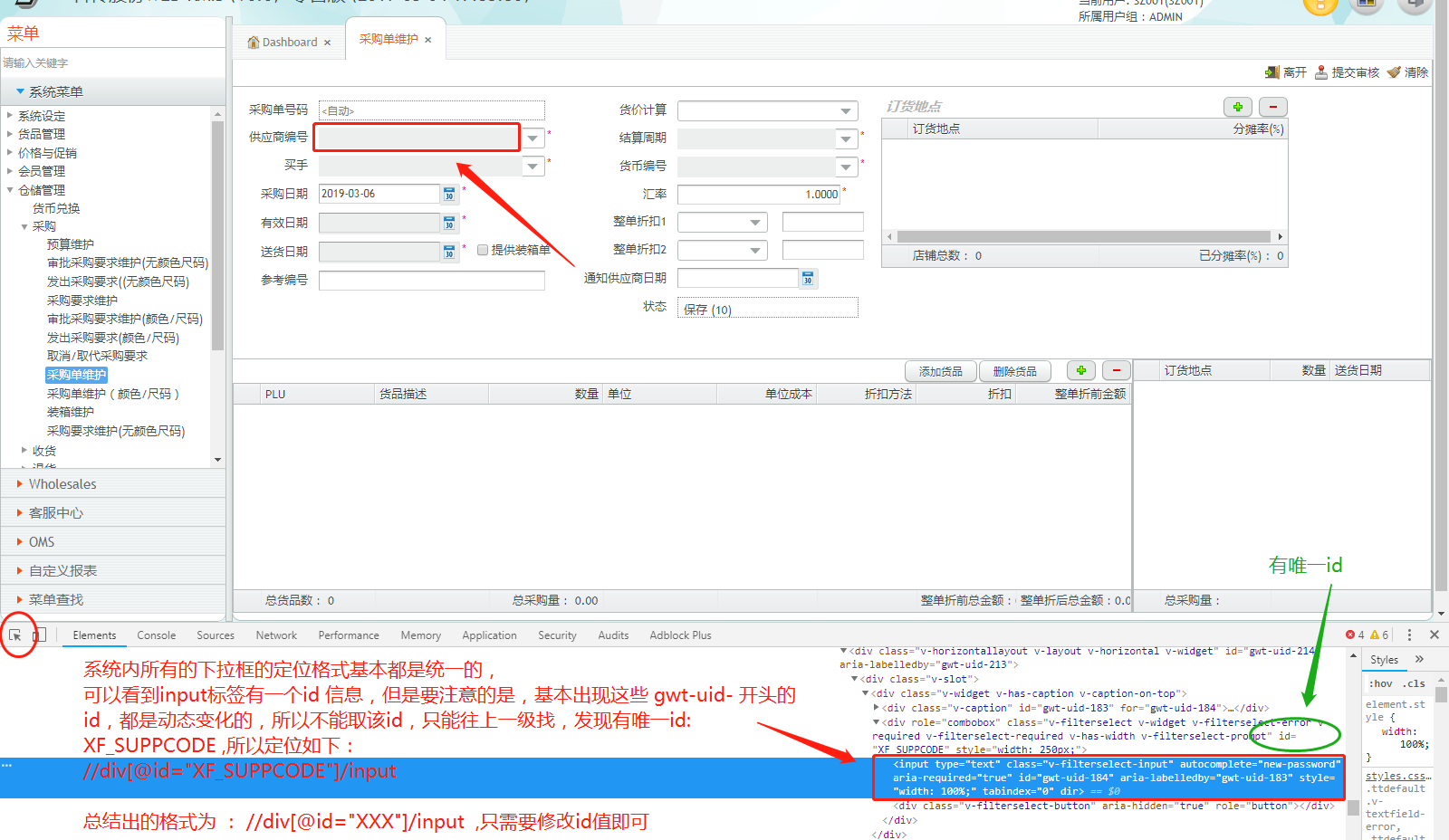
例如以上例子： //span[text()="采购单维护"]

1）、text() 的文本必须是和span标签里面的文本**完全匹配**，而且如果**有空格也需要保留空格**，所以建议直接将span里面的文本直接copy，减少出错的概率

2）、文本定位的格式基本都比较固定， //标签名[text()="XXX"] , "XXX" 表示 标签里包含的文本

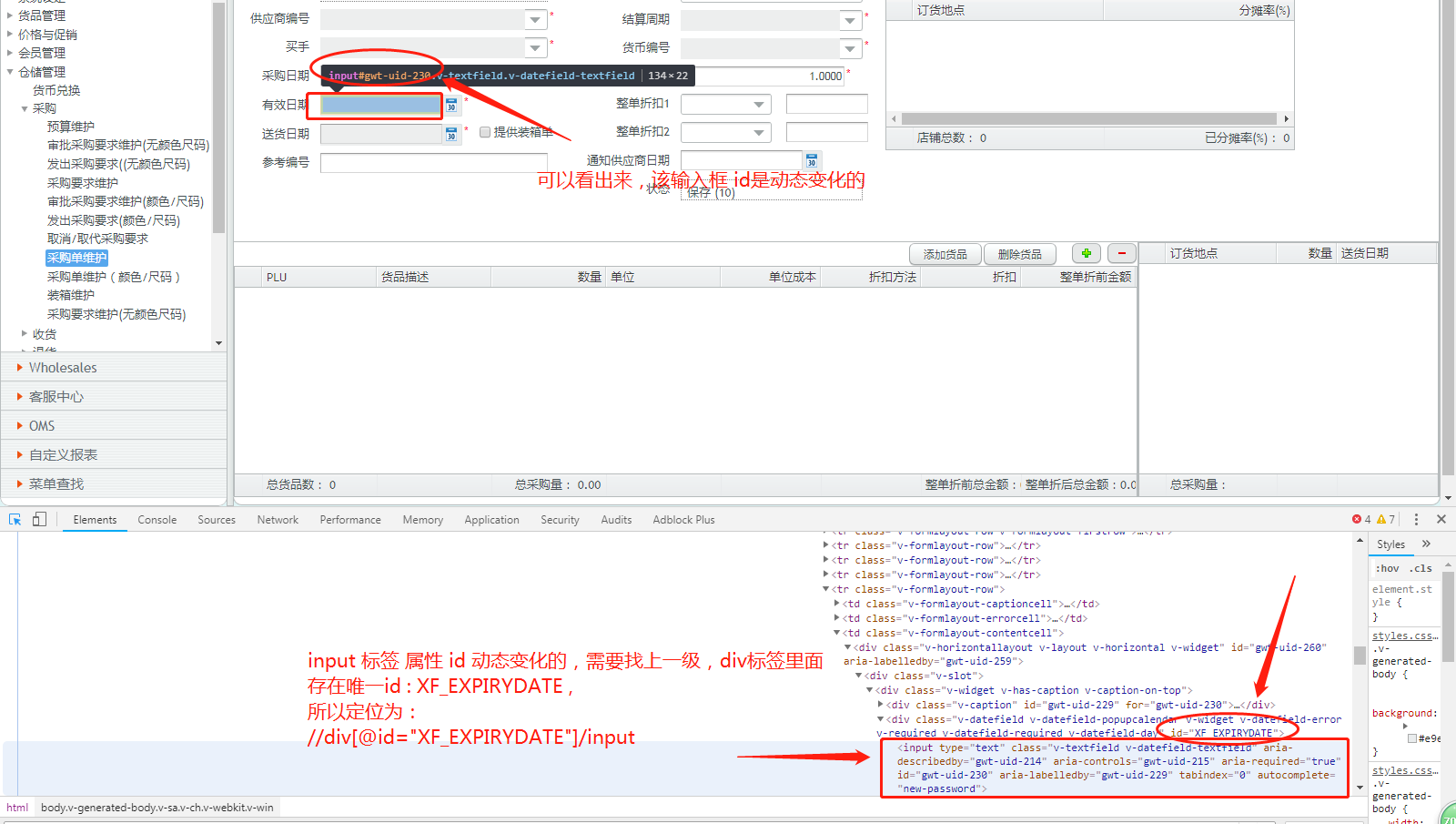
###### C、定位下拉框

定位下拉框都需要借助它的上一级进行间接定位，如下图：



###### D、定位日期输入框

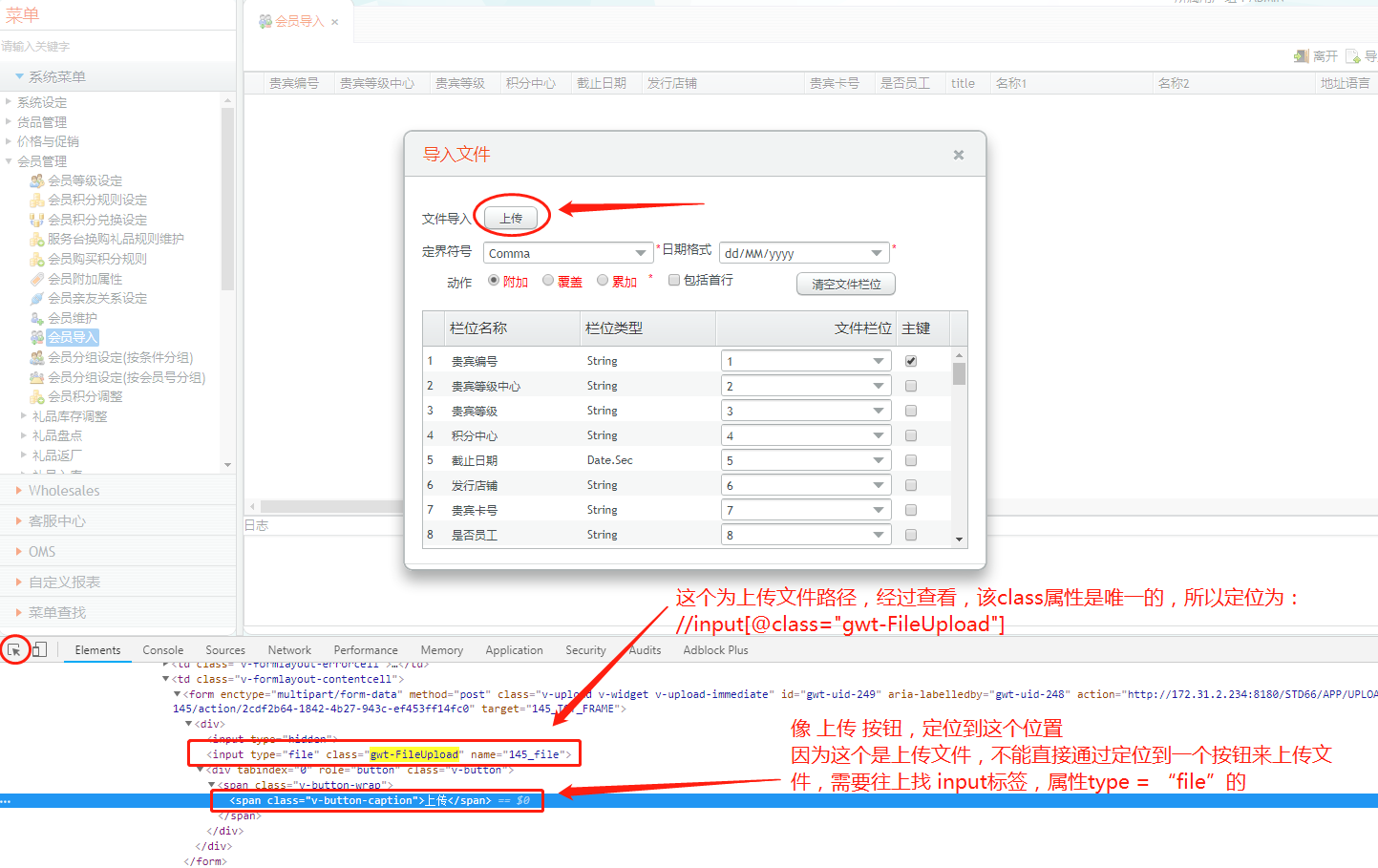
定位日期输入框和下拉框方法一致



###### E、定位文件上传按钮

**特别注意：上传按钮的定位和其他元素的定位不太一样**

上传文件按钮最终是为了上传文件的，所以不是要定位上传按钮

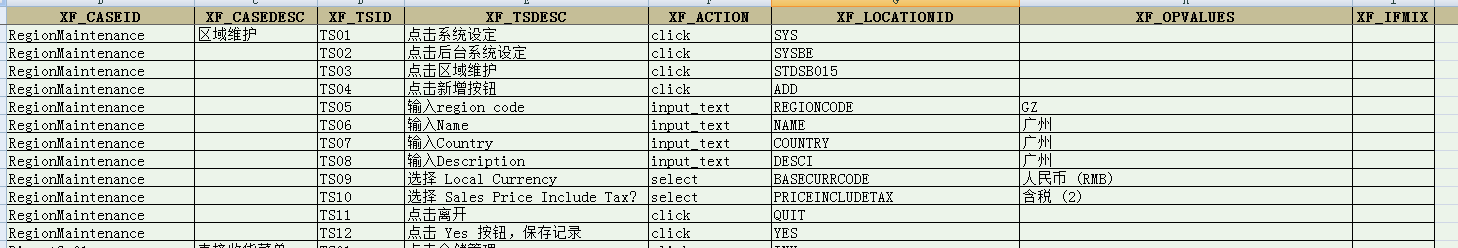


### 2.2、维护测试用例

#### 2.2.1用例表介绍

所有的测试用例信息都维护在表XF\_TESTCASE，要执行用例，XF\_TESTCASE表就必须要有维护用例信息，简单来说该用例表里面保存着对浏览器的一切操作

表XF\_TESTCASE表如下图所示：



以上字段解释如下：

**XF\_CASEID :** 该id是每个用例的唯一id，由个人自定义，建议根据测试模块，起一些有意义的id，例如测试 区域维护模块，可以起这些id ：RegionMaintenance\_01, RegionMaintenance\_02 ....

**XF\_CASEDESC :** 描述用例

**XF\_TSID :** 用例测试步骤id，需要按顺序递增，项目执行的时候会按照测试步骤id从小到大往下执行，建议id 以TS开头， 如：TS01，TS02 ......

**XF\_TSDESC :** 测试步骤描述，最好每一个测试步骤要有一个详细的描述，例如：点击新增按钮

**XF\_ACTION :** 这个是关键字，这个的作用就是对浏览器的操作

**XF\_LOCATIONID :** 该id为 表XF\_PAGELOCATION 中的XF\_LOCATIONID

**XF\_OPVALUES :** 该值为关键字对应的参数值

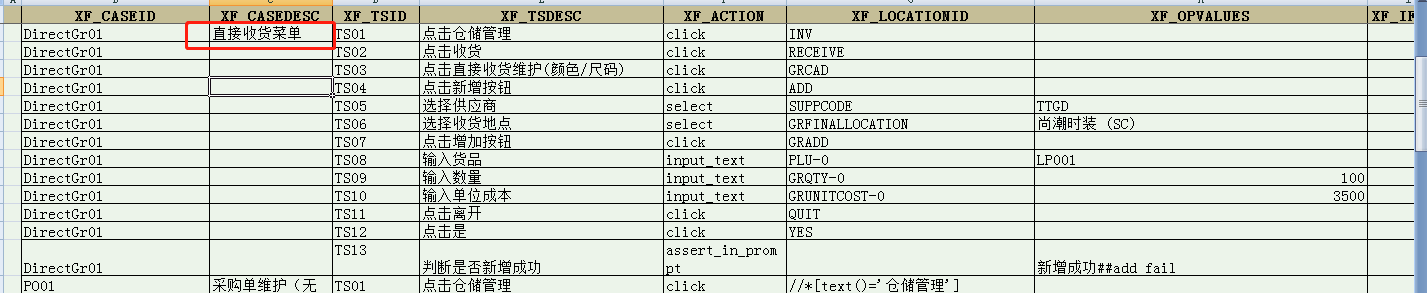
**XF\_IFMIX :** **解释后补**

#### 2.2.2写用例指引

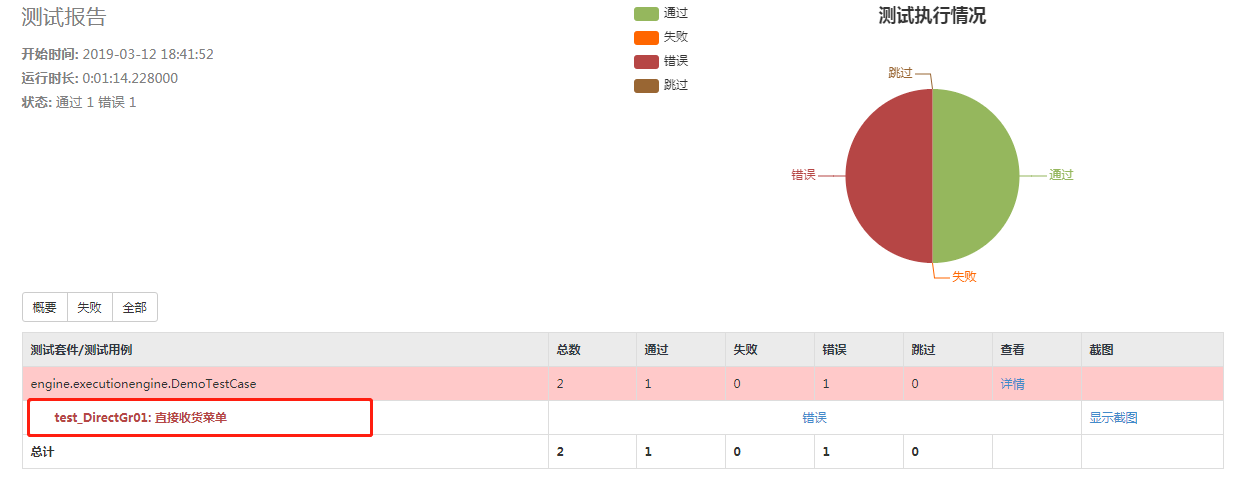
##### 1）、用例描述

写每一个用例之前需要给每一个用例增加描述，以便在出测试报告的时候可以更加容易阅读。

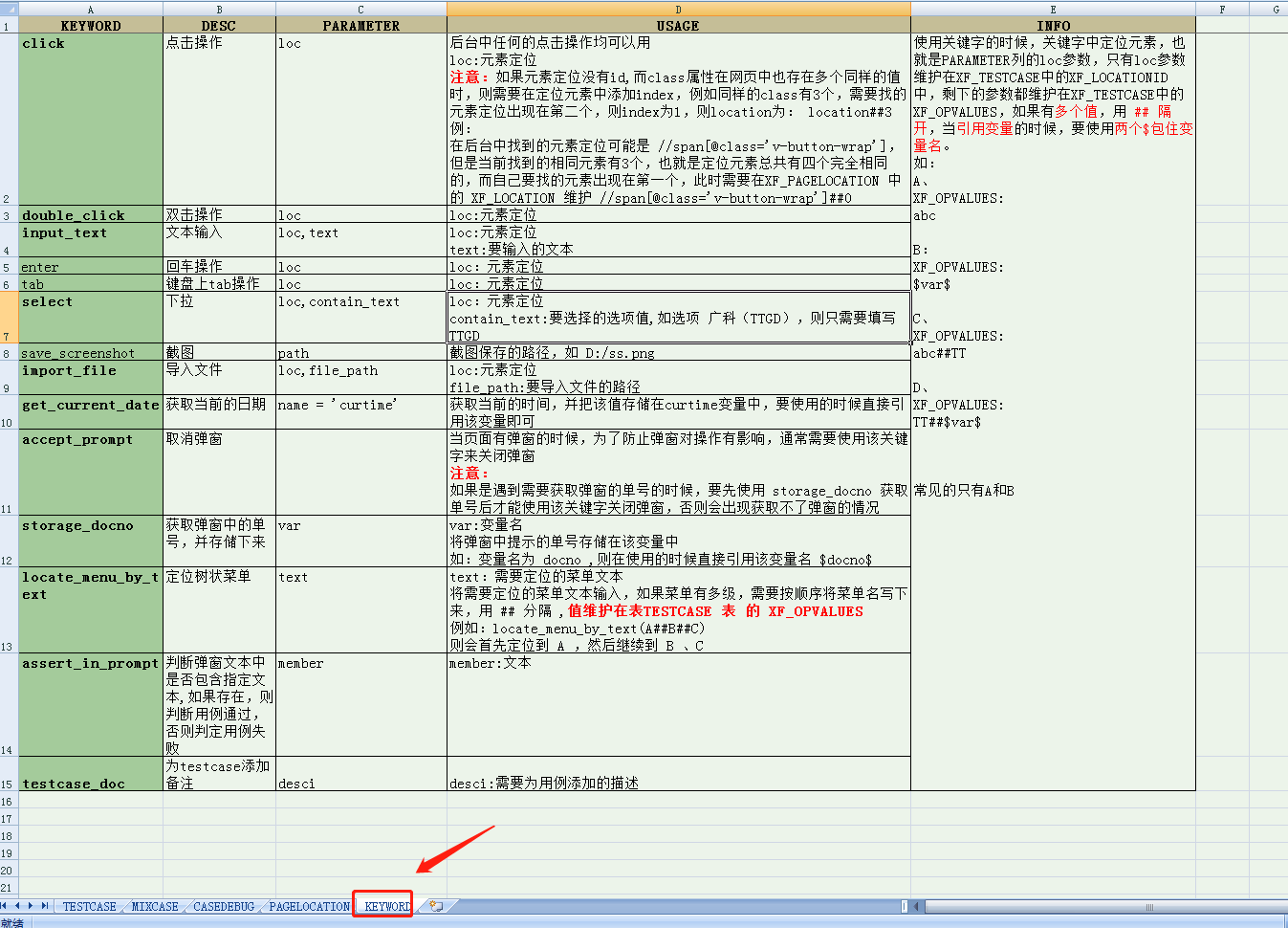
只需要在每一个case的XF\_CASEDESC列的第一行维护，在其他行维护无法将该描述添加到report



最终的效果如下：



##### 2）、选择关键字和填写关键字值



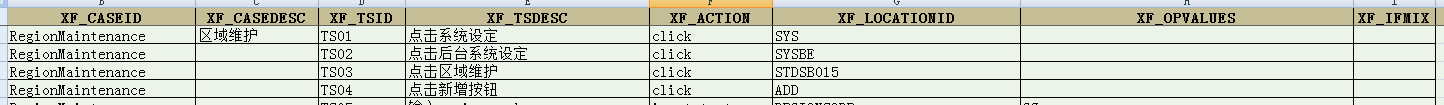
以上关键字解释均可以在EXCEL 中 sheet KEYWORD中查找

##### 3）如何查看上表和填写XF\_TESTCASE表

A、查看上图列DESC的解释，选择适合的关键字

B、PARAMETER 列：也就是参数列

**参数loc才填写到 列XF\_LOCATIONID,其他的都填写到XF\_OPVALUES**

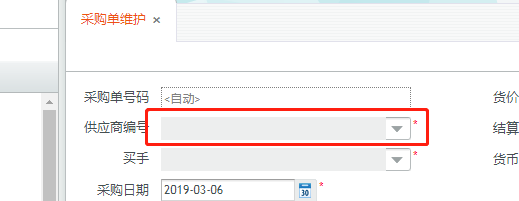


例如上图XF\_ACTION列为关键字，click 关键字中的参数只有loc，所以只需要在XF\_LOCATIONID列中填上值

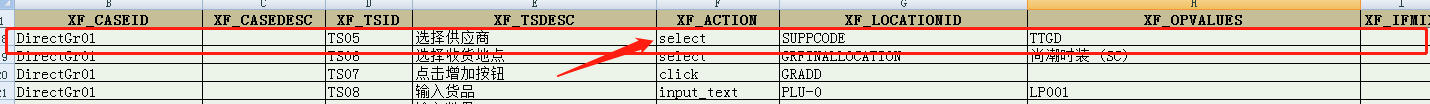
##### 4）关键字应用场景

select、import\_file、get\_current\_date、accept\_prompt、storage\_docno、locate\_menu\_by\_text、assert\_in\_prompt、testcase\_doc

**select :**



像这类下拉框，使用select关键字





至于XF\_OPVALUES的填写，如上图，如果需要选择 广科（TTGD） ，则XF\_OPVALUES填写TTGD即可，或者填写全部也可以

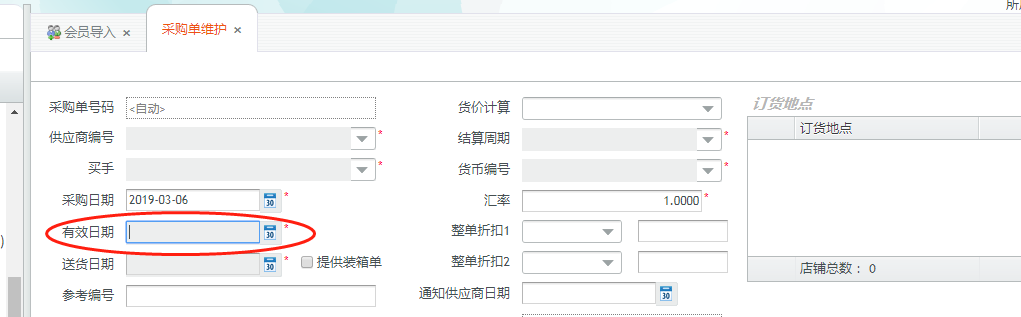
**import\_file:（需要补充明细）**

如下图，需要导入文件

****

**get\_current\_date:**

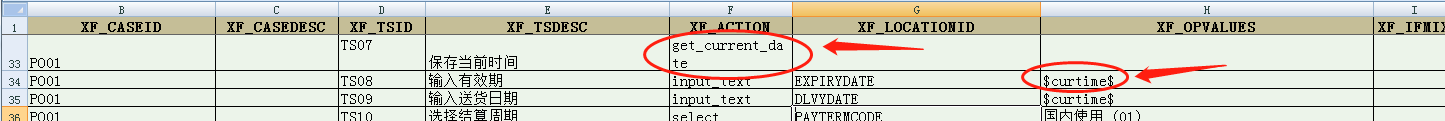
比如：填写下面的有效期，需要自动获取当前日期，就需要用到get\_current\_date 关键字



get\_current\_date 获取到的时间会存储到变量 curtime 中

需要用到该变量的时候，再引用，用法如下：

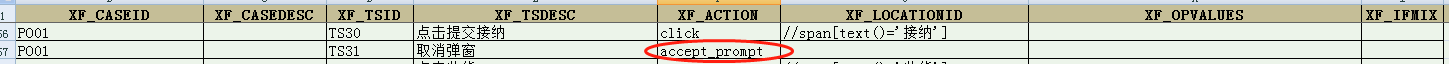
需要先使用关键字get\_current\_date 获取当前时间，然后引用的时候使用 $curtime$



**accept\_prompt:**



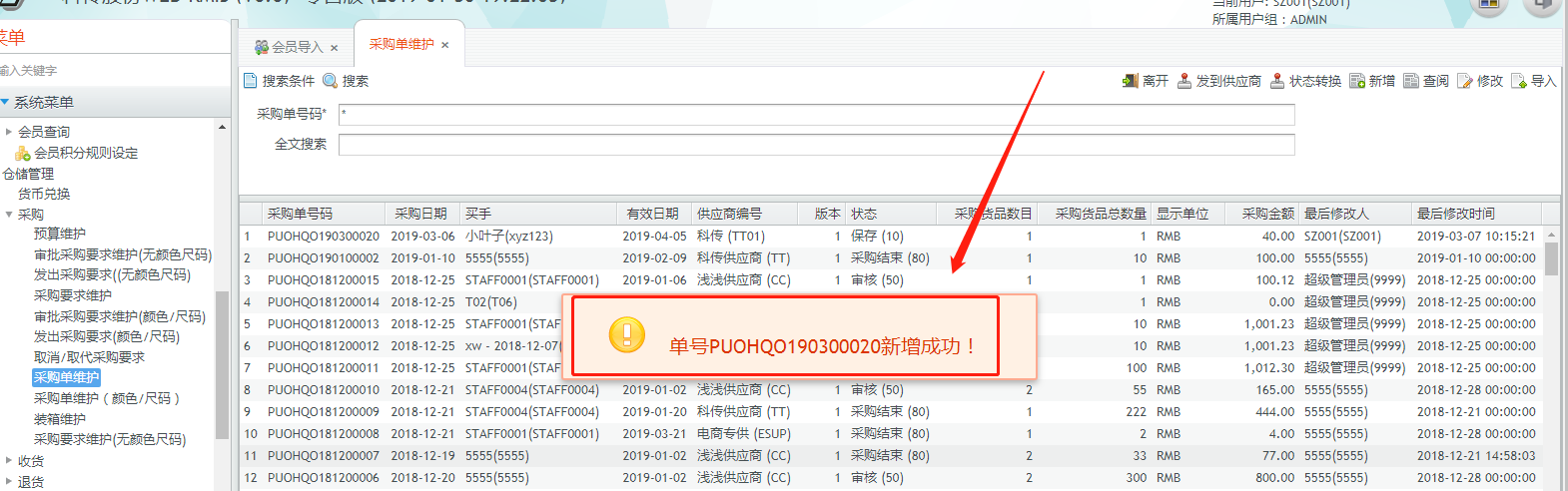
例如，需要取消以上弹框，可以直接使用accept\_prompt 关键字



在页面弹窗影响到下一步操作的时候，可以通过accept\_prompt 关键字来取消弹窗

**storage\_docno:**

该关键字主要是用来获取**屏幕弹框的单号**，并且存储下来



**注意：要使用storage\_docno 关键字获取单号，那上一个步骤一定不能使用accept\_prompt关键字来取消弹窗，否则无法获取单据号**

以下提供该关键字的一些**常用场景**：

1、新增一个单据后，重新进入该单据修改单据内容

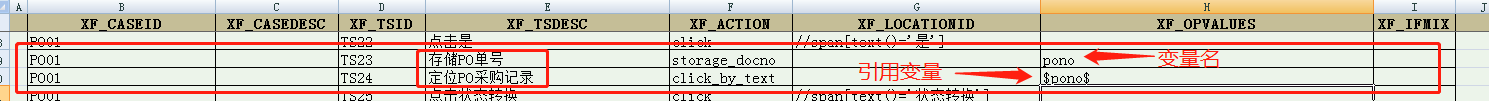
例如：

新增一个采购单据，因为在生成采购单之前，我们无法得知单据号，所以要在生成采购单据记录的时候，想要定位该记录是做不到的，如图下所示：



上图的记录定位只能通过文本定位该记录，但是上面的采购单号码是只有在生成采购单后才能得知，这时候就可以使用storage\_docno 存储单号到一个变量里面，然后再引用该变量来定位采购单记录。

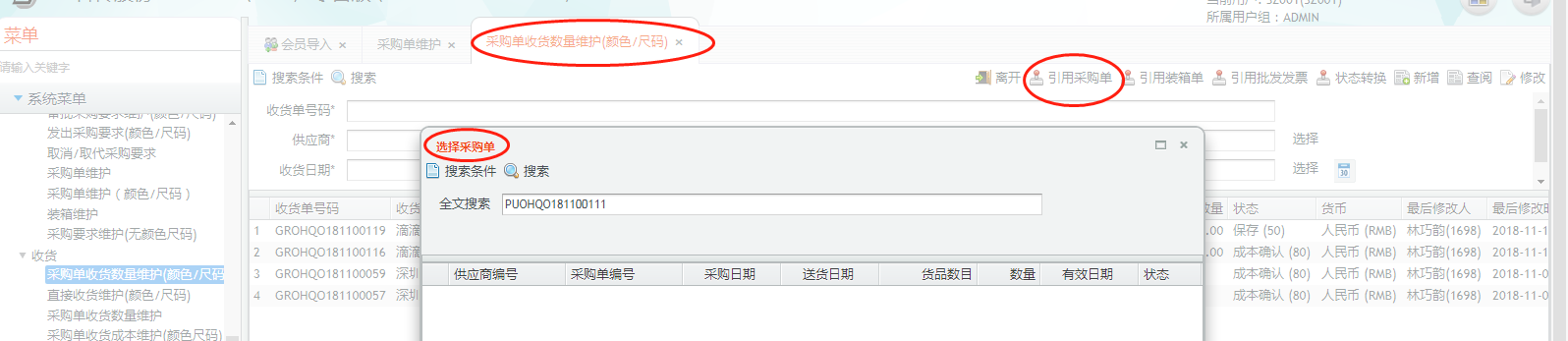
写testcase时可以这么做：



2、新建两个单据，下一个单据B需要引用上一个单据A

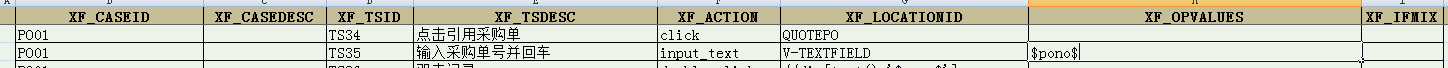
例如：

通过采购单来收货



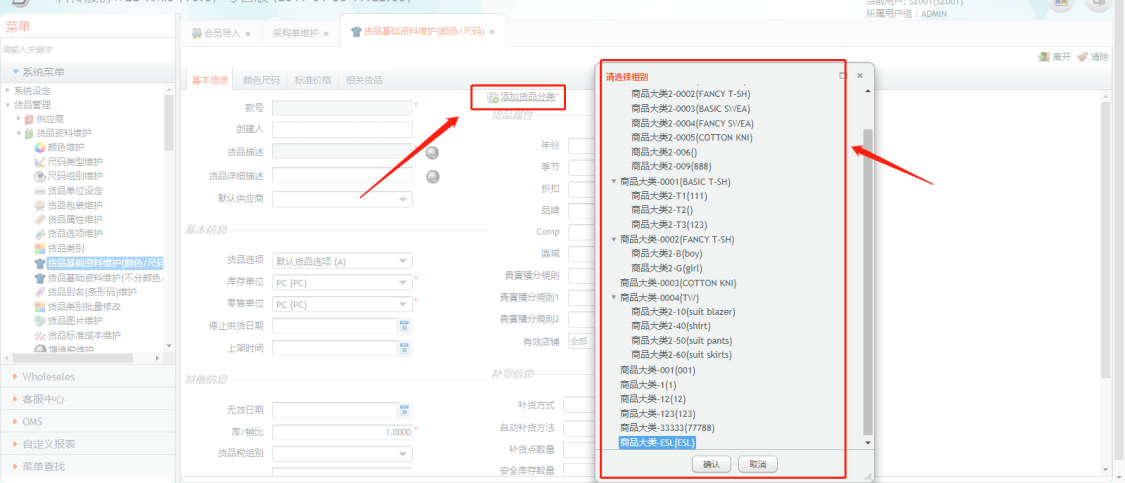
因为无论新建任何单据，在写用例的时候都无法获知单据号，所以必须得通过 storage\_docno关键字来获取采购单号并且存储下来，以便再次引用。

上一步存储单号到 pono变量中，再次引用可以这样做：



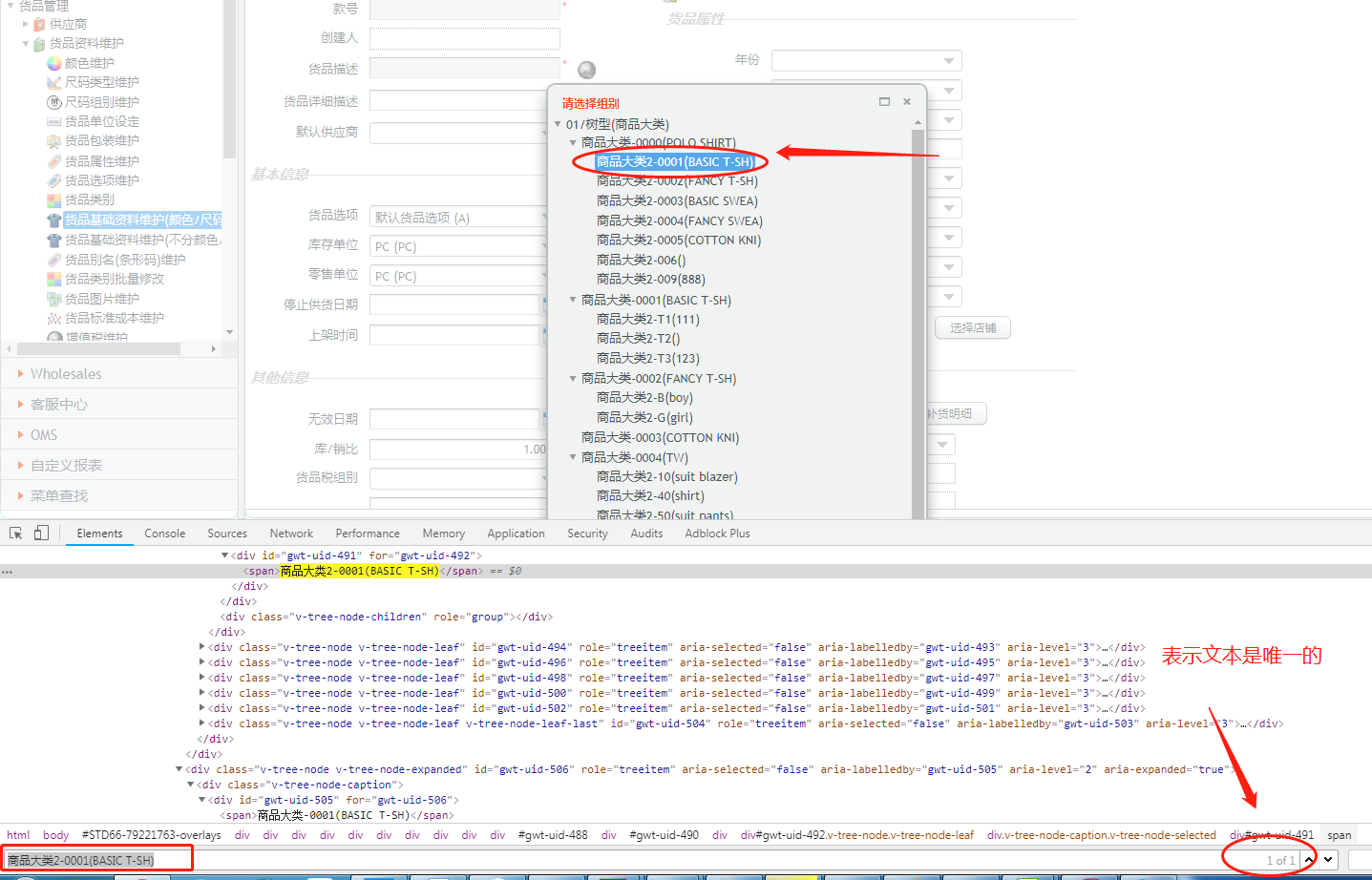
**locate\_menu\_by\_text:**

该关键字用来定位下面的菜单，该关键字是通过文本定位

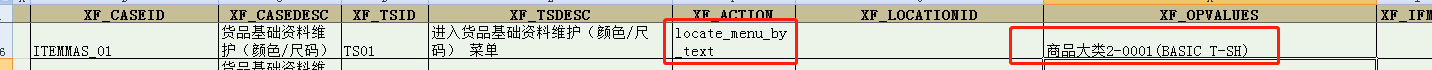


locate\_menu\_by\_text 能够模糊匹配，只要是搜索该文本没有重复的，都可以

以这个为例子举例：

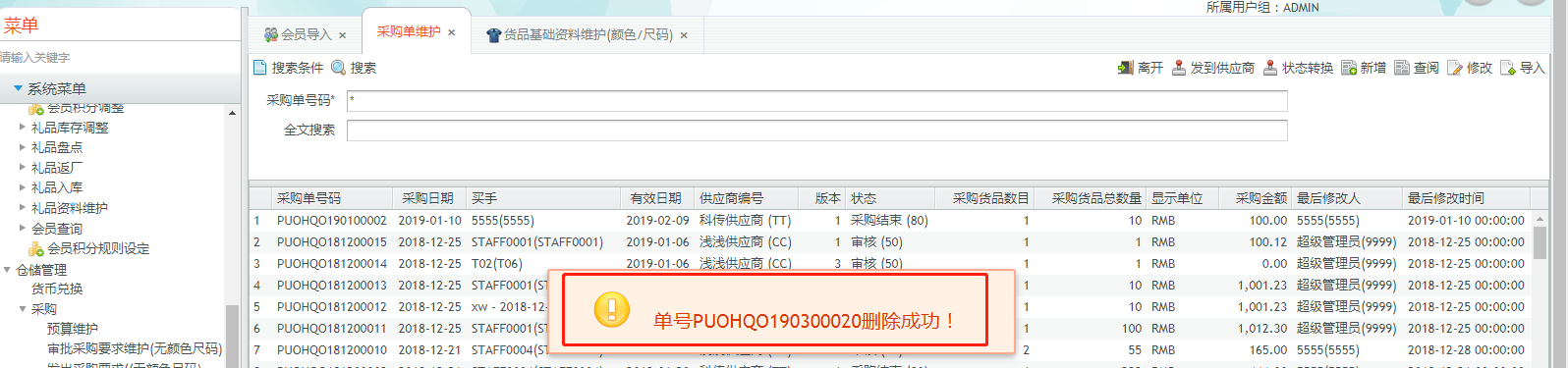


需要定位上面的菜单，可以这么写

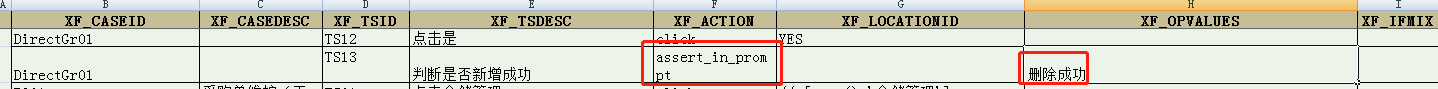


**assert\_in\_prompt:**

该关键字只能用来断言弹框的文本



例如，我们删除一张采购单的时候，要判断是否删除成功可以通过弹框文本断言



这个表示弹框中如果包含 “删除成功” ，表示断言成功，也就表示testcase 执行成功

**testcase\_doc:**

该关键字用来描述测试用例，主要是为了让report更加易读



如下图：采用testcase\_doc的效果



### 2.3执行测试用例

维护好测试用例后，最终还是要执行用例才能得到测试报告

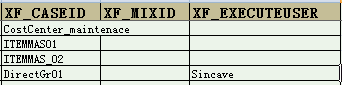
#### 2.3.1调试部分用例

这个功能主要是用于调试一个或者是某些用例。例如，刚添加一个用例 AAA，想要试试用例AAA是否能否成功执行，就可以用这个功能。配置如下：

#调试用例，mode = 0 表示执行用例表里面的全部用例，mode = 1，表示执行指定的用例，executeuser 表示执行指定的user用例，如果为空，表示执行没有指定user的用例**[Debug]  
Mode** = **1  
Execute\_User** = **xxx**

在config.ini 里，将上方的 Mode 修改为 1 ，即可切换到调试用例模式下

以下表为XF\_CASEDEBUG 为调试用例表，需要调试的用例都需要放到这个表



如上图所示，只需要将要执行的用例id或者组合用例id添加到上方，XF\_CASEID和XF\_MIXID只能选填一个。（组合用例在 2.5组合测试用例 详细介绍）

另外，根据情况指定执行用户 Execute\_User：

**A、例如，Execute\_User = Sincave ，那执行用例的时候，只会执行 XF\_EXECUTEUSER中值为Sincave的用例**

**B、如果 Execute\_User = ，也就是不指定执行用户，那就会执行 XF\_EXECUTEUSER中值为空的用例**

做完以上所有的设定以后，需要运行 AutoTestV66\runall\run.bat 文件来执行用例

#### 2.3.2执行全部用例（后补）

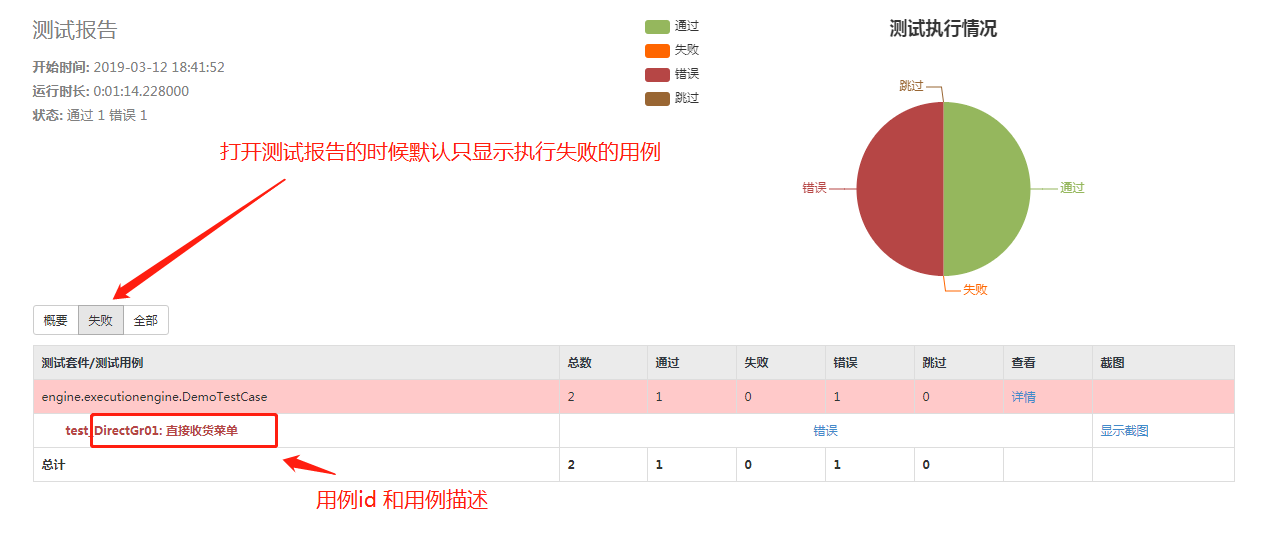
如果已经调试好所有的用例以后，就需要切换到普通模式来执行全部用例

#调试用例，mode = 0 表示执行用例表里面的全部用例，mode = 1，表示执行指定的用例，executeuser 表示执行指定的user用例，如果为空，表示执行没有指定user的用例**[Debug]  
Mode** = **0  
Execute\_User** = **xxx**

可以通过修改Mode = 0 来切换到普通模式，切换到普通模式后，Execute\_User 配置失效。普通模式下会执行

### 2.4查看测试报告

测试报告在该路径下 AutoTestV66\report\htmlreport 查找



### 2.5组合测试用例（后补）